

AMENDMENT OF CLAIMS

Claim 1 (amended):

A3
Sub C2

1. A method for operating three-dimensional (3D) application software to generate a 3D stereoscopic vision display, wherein the 3D application software is of the type that intended to provides a 3D output signal to a display driver for a 3D graphics rendering device to generate a display output to appearing to be three-dimensional for display on a two-dimensional (2D) screen display, comprising:

(a) running the application software in its normal mode to generate the 3D application data output signal which is normally to be sent from the application software to a to an application programming interface (API) display driver for a 3D graphics rendering device for generating a 3D display output on a the 2D screen display;

(b) intercepting the 3D application data output signal from the application software and redirecting the data it to a pseudo driver for generating a 3D stereoscopic display, wherein said pseudo driver generates from said output signal a left image view signal and a right image view signal that is stereoscopically offset from left image view signal; and

(c) using the pseudo 3D display driver to generate a 3D stereoscopic display- said pseudo driver providing said left image view signal to a first 3D graphics rendering device and said right image view signal to a second 3D graphics rendering device, and using said first and second 3D graphics rendering devices for separately rendering in tandem left and right image views for display in a 3D stereoscopic vision display device.

Claim 2 (amended):

2. A method according to Claim 1, wherein the 3D stereoscopic vision display device is selected from the group consisting of head-mounted "stereo vision" goggles, head-mounted 3D display device, and a stereo vision monitor.

Claim 3 (original):

3. A method according to Claim 1, wherein the 3D application software is a 3D video game software which provides 3D game data output.

Claim 4 (amended):

4. A method according to Claim 3, wherein the intercepting and redirecting of the 3D game data is obtained by providing a wrapper for the game software's native API display driver ~~having and replacing with stereoscopic pseudo driver~~ display function calls linked under the same name as the game software's native API display driver for a 2D display.

Claim 5 (original):

5. A method according to Claim 4, wherein the wrapper supports a selected one of the following group of native API formats: Glide; OpenGL; and DirectX.

Claim 6 (canceled)

Claim 7 (amended):

7. A method according to Claim 1, wherein the pseudo driver generates a 3D stereoscopic vision display using one physical graphics card with dual graphics generator card heads for separately rendering right and left image viewpoints for the 3D stereoscopic vision display.

Claim 8 (original):

8. A method according to Claim 3, wherein the intercepted 3D game data is stored in a 3D data recorder for later play back.

Claim 9 (original):

9. A method according to Claim 8, wherein the recorded 3D game data are transmitted or downloaded through an online interface to a remote user.

Claim 10 (original):

10. A method according to Claim 3, wherein the intercepted 3D game data is combined with other 3D content using a mixer and a dual rendering system.

Claim 11 (original):

11. A method according to Claim 10, wherein the dual rendering system is kept running while switching between different game software.

Claim 12 (original):

12. A method according to Claim 3, wherein another pseudo driver operates on the 3D game data in tandem with the pseudo 3D display driver.

Claim 13 (original):

13. A method according to Claim 12, wherein the other pseudo driver is a stereo sound or a directional force feedback driver.

Claim 14 (original):

14. A method according to Claim 12, wherein the video game software is run with one or more tracking devices for input from the player.

Claim 15 (amended):

15. A method of generating a 3D multi-view display system for operating operable with three-dimensional (3D) application software of the type that provides a 3D output signal from the application software to a display driver for a 3D graphics rendering device to generate a display output appearing to be three-dimensional on a two-dimensional (2D) screen display and which makes display function calls to a native API display driver for the software under an API linking name to provide a the 3D display output to a two-dimensional (2D) screen display, comprising:

(a) a computer for running the application software on a computer in its normal mode to generate a 3D application data output signal intended for a 2D screen display;

(b) a file directory system for the computer in which the application software's native API is normally stored under the API linking name; and

(c) providing a pseudo 3D display driver stored in the computer's file directory system under the API linking name as a wrapper in place of that links to the native API display driver by for intercepting the image display function calls to the native API display driver and 3D

Cont
A3
~~application data output from the application software and redirecting them through the pseudo 3D display driver in order to generate multiple, separate image views and provide them to respective ones of a corresponding multiple of 3D graphics rendering devices for a multi-view 3D stereoscopic display.~~

Claim 16 (amended):

16. A 3D multi-view display system method according to Claim 15, wherein the 3D ~~stereoscopic~~ multi-view display is selected from the group consisting of head-mounted "stereo vision" goggles, head-mounted 3D display device, and a stereo vision monitor.

Claim 17 (amended):

17. A 3D multi-view display system method according to Claim 15, wherein the 3D application software is a 3D video game software which provides 3D game data output.

Claim 18 (amended):

18. A 3D multi-view display system method according to Claim 17, wherein the wrapper supports a selected one of the following group of native API formats: Glide; OpenGL; and DirectX.

Claim 19 (amended):

19. A 3D multi-view display system method according to Claim 15, wherein the pseudo 3D display driver ~~specifies~~ generates right and left eye image views for the 3D application data output, and sets up provides them to respective right and left graphics rendering devices in parallel rendering engines using the native API for converting the right and left eye image views into right and left image data display outputs, respectively, which are used for the a 3D stereoscopic vision display.

Claim 20 (amended):

20. A 3D multi-view display system method according to Claim 19, further including separate graphics generator cards for rendering the right and left image displays views in parallel for the 3D stereoscopic vision display.